

BENCHMARKS

VOLUME 2 NUMBER ⁴~~3~~

NORTH TEXAS STATE UNIVERSITY

JUNE 1981

*BENCHMARKS was produced using MUSIC/SCRIPT

TABLE OF CONTENTS

1.	CONSULTING POLICY	1
2.	OPERATIONS	2
	Summer Computing Hours	2
	Priority Setting For Batch Jobs	2
3.	NEW "UTILITY" PROCEDURE	3
4.	MUSIC	4
	Electronic Mail System Revisited	4
	Capturing Output From OSJR	6
	Additional CAI Available Through MUSIC	6
	Future MUSIC Software	10
	MUSIC Notes	10
5.	HP-2000	11
	Getting Started On The HP	11
	The Big Purge	12
	ISB 224 No Longer A Terminal Room	13
	Summer Backup Schedule	13
6.	SAS PROCEEDINGS AVAILABLE FOR EXAMINATION	13
7.	RESULTS OF THE WORKSHOP/SEMINAR POLL	13
8.	INDEX TO PAST ISSUES	15
9.	CALLING ALL COURSES USING STATISTICAL PACKAGES	17

CONSULTING POLICY

The consulting policy of Academic Computing is currently undergoing some review within The Computing Center. In the very near future we intend to take a recommended policy to the Computing Council. If you have any thoughts concerning the nature and extent of consulting offered by Academic Computing, please put those thoughts in writing and address them to: Tom Madron, Manager of Academic Computing, NT Box 13495. Alternatively, you may send mail to "the system" - see "Electronic Mail System Revisited" in this issue for further information. Your help will be appreciated.

* * * * *

* OPERATIONS *

* * * * *

SUMMER COMPUTING HOURS

Computing hours for both the summer sessions are essentially the same as the hours maintained during the spring semester. The following hours will be maintained throughout the semester or until further notice:

Computing Center RJE: 24 hours a day Monday through Friday, close Saturday midnight. Noon until midnight on Sunday.

Computing Center Computer Room: Same hours as Computing Center RJE. (used primarily for jobs requiring special handling).

Business Administration Building
Computing Access Facility : 8:00 AM to Midnight Monday through Saturday. Noon to Midnight Sunday.

PRIORITY SETTING FOR BATCH JOBS

Every job processed by the MVT operating system is assigned an execution priority and a print priority. The execution priority is assigned when the job enters the system and is a number from 0 to 8, based on the time and the number of lines estimated in the JOB card (the April, 1981, BENCHMARKS addresses the topics of JOB card preparation and job processing). For the job classes currently being serviced, the job with the highest execution priority will be selected to run next. When more than one job has the same priority, the order in which the jobs entered the system determines which job is executed first.

In general, the less time and lines estimated in the JOB card, the higher execution priority the job receives and the sooner the job will be executed. This means that the more accurately a user can estimate time and lines, rather than putting "safe" large numbers in these parameters, the better turnaround (s)he will get. Underestimating, of course, is potentially more time consuming than overestimating, since the user gets a job that has abended and must resubmit it.

After the job has executed, the number of lines actually generated is used to determine the print priority - a number from 0 to 8. The fewer lines generated the sooner the printed output will be available for distribution. The actual value can be computed only after the job has run.

Obviously, jobs with large time and/or lines estimates could rest on the input queue for an indefinite period of time. To offset this possibility, jobs are "priority aged" approximately every 30 to 45 minutes (i.e., the priority is increased by 1) that they remain on queue, until they are processed or the priority reaches 12. This guarantees that the job will not remain in the system indefinitely.

Prior to the acquisition of the AS/5000 and multiple operating systems (VM, MUSIC, and others), the priority was based on "real time"

(i.e., actual clock time elapsed), and aging took place exactly every 30 > 45 minutes. Currently, however, priority aging takes place using "virtual time" (the time that MVT is using the CPU - see BENCHMARKS June, 1980, for a brief discussion of the concept of "virtual machines"). Thus, priority aging depends on the amount of activity on the system as a whole. Heavy MUSIC usage, for example, will cause priorities to be aged more slowly (in real time) because MVT is sharing the system with MUSIC, and its timer is only activated when it is actually running.

The execution priority for a job may be determined from the following table by adding the priority number to the right of the estimated time (or next higher time if between two times) to the priority number to the right of the estimated lines (or next higher value if between two lines values) and then subtracting 8. If the resulting number is negative, the priority is 0. The print priority may be found directly from the right side of the table, using the actual number of lines printed.

<u>ESTIMATED TIME m:s</u>	<u>PRIORITY NUMBER</u>	<u>ESTIMATED LINES (1000's)</u>	<u>PRIORITY NUMBER</u>	<u>ACTUAL LINES</u>
:05	8	1	8	500
:10	7	2	7	1000
:15	6	3	6	2000
:30	5	5	5	3000
1:00	4	10	4	5000
2:00	3	20	3	10000
3:00	2	45	2	20000
5:00	1	>45	1	45000
>5:00	0		0	>45000

Example: If the time estimate is 30 seconds and the line estimate is 7 (i.e., 7,000 lines), the execution priority would be $5+4-8=1$. If the job actually generated 4000 lines, the print priority would be 4.

NEW "UTILITY" PROCEDURE

A new procedure, called "UTILITY", has been added to the procedure library which makes the use of IBM's utility, IEBGENER, much easier. The UTILITY procedure provides a means for copying datasets with a minimum of job control language (JCL) through the use of parameters on the EXEC card. The procedure is invoked by:

```
// EXEC UTILITY
```

and the possible parameters are as follows:

,DATAIN='name': the dataset name (DSN) for a disk file or a file on a standard labeled tape.

,DATAOUT='name': The output DSN if output is to be on a standard labeled tape or on disk.

If the dataset is to be output to disk or tape, then the following parameters must be used:

,MODE=FB: Record format of the output dataset, default=FB (fixed, blocked).

,RECL=80: Logical record length of data, default=80.

,BLOCK=3520: Blocksize of data, default=3520.

,PRIM=5: Primary space allocation, default=5 tracks.

,SEC=5: Secondary space allocation, default=5 tracks.

,VOLOUT='volume serial name': Volume name of disk or tape for output file. The default is nothing which, for disk datasets, allows the system to find the space. For most situations, the output dataset should be placed on ACAD00, ACAD01, or ACAD02. This parameter must always be coded with tape volumes.

,UNITOUT=SYSDA: Output device, default=SYSDA. This should not be changed for an output disk file. For an output tape file it should be coded TAPE9.

If a dataset is being output to tape, it is also necessary to provide one short DD statement with the ddname "NO" with the following elements:

```
//NO DD LABEL=n
```

where n is the sequential number of the dataset on the tape (this is for a standard labeled tape).

If it is necessary to print a dataset, UTILITY can be used by simply coding PRINT=YES along with any information regarding the input dataset. If the input dataset is on cards, then code the following card and place it before the data and after the EXEC card:

```
//SYSUT1 DD * (or if the dataset contains JCL, then substitute the word DATA for the asterisk (*)).
```

```
* * * * *
*           M U S I C           *
* * * * *
```

ELECTRONIC MAIL SYSTEM REVISITED

As reported in the April BENCHMARKS, an electronic mail system has been installed and can be accessed through MUSIC. Since the installation of the mail system and its documentation in the newsletter, however, some changes in the system have taken place. Entering: HELP MAIL will give a potential mail user information much like the following (it should be noted that the HELP files are updated whenever there are changes that need to be documented - thus it would be wise to execute HELP whenever you are in a situation that seems unfamiliar or confusing to you - only within MUSIC, of course):

To build a mailbox simply type:

```
MAILBOX.BUILD
```

Both the sender and the receiver of mail must have a mailbox "built"

before (s)he can receive or check receipt of mail.

This command creates two files in the user's save library:

1. code.MAILBOX (where "code" is the first four characters of the user's ID code; this is the "send" file).
2. @MAIL (this is the "receive" file).

To send mail type:

MAIL NOACK, CODE, LETTER

NOACK is optional. It turns off the feature that ACKnowledges the sender that his/her mail has been received and read.

CODE can be defined as follows:

- 1) NNNN, which is the first 4 characters of the receiver's ID code.
- 2) (xxxx,yyyy,zzzz), used to send mail to more than one user (receiver).
- 3) @NN, where NN is a save library file which contains the first 4 characters of all the ID's who are to receive the letter (this file may have any name following the same naming conventions that apply to any other save library file). Each record of the file must contain ONE IDCODE starting at column 1. For example:

XXnn
 YYnn
 ZZnn
- 4) *, Which indicates the letter is a MEMO to the sender (i.e. to yourself). The user will be prompted for the date that (s)he wants to receive the MEMO the date of the following forms:
 - a) yyddd - (e.g. 81231)
 - b) mm/dd/yy - (e.g. 12/31/81)
 - c) *+nnn - this is the relative date which means today's date plus nnn days - (e.g. *+12)
- 5) SYSTEM, which will send the letter to the system and will then be read by the MUSIC coordinator(s). This "system" mail will be acted upon in a manner which is deemed appropriate by the MUSIC coordinator(s). They will probably either (a) reply to the sender via the mail system or (b) forward the mail to someone else who will then probably reply to the sender via the mail system.

LETTER is the name of the file to be sent. An * may be entered for direct input of the message.

To read your mail type: RDMAIL (note: this is new). Each letter will be described by the sender and the date/time. If you want to read the letter press the enter/return/carriage return key. If you want to throw the letter out type: JUNK. If you want to keep the letter, but not read it at the moment, type: KEEP.

Additionally, a new parameter, SCAN, can be used with RDMAIL. This causes a brief list of letters waiting in your mailbox to be displayed. To use this feature simply enter RDMAIL SCAN when your terminal is in *GO mode.

RDMAIL can be executed automatically each time you sign-on to MUSIC (and/or for MEMO's). This is accomplished by setting the AUTO parameter in your user PROFILE. To do this, enter: PROFILE AUTO=RDMAIL. In order for this to work you must have already created a mailbox under your ID code.

It should be apparent that the electronic mail system has the potential for being quite a handy tool for MUSIC users. With a little

imagination, one can think of all kinds of nifty things that MAIL can do: (e.g) send suggestions and questions to Computing Center personnel; communicate with the students in your class; communicate with your professor; carry on a lively discourse with a friend; and remind yourself of important matters. Send us some mail and tell us how you like it!

CAPTURING OUTPUT FROM OSJR

A handy feature of the FILE statement in OSJR that may not be readily apparent is the ability to copy output from OS to a MUSIC save file. For example, if you have some data that you need to analyze using one of the North Texas Statistical procedures, but that needs to be "massaged" a bit before it is in the proper form for the NT procedure you are planning to use, you could:

1. Execute, through MUSIC, an SPSS, SAS, or other program that collapses, computes, or whatever you need to do to your data to get it ready for the NT procedure and then writes the transformed data to the card punch.
2. Examine your output through OSJR, making sure that everything worked as planned. When you are sure that your efforts were successful, enter: OUT DSID=4,file=nnnnn where nnnnn is the name of a SAVE file that you want your card image output to go to. You can then PURGE your output from OSJR. (NOTE: For more information about DSID, FILE and other parameters in OSJR, execute OSJR HELP and/or see page 50 in the MUSIC at NTSU Manual - it is possible to use the UNIT parameter to capture output also, but the FILE parameter is easier to use).
3. Execute, through MUSIC, the North Texas procedure you wanted to run, noting that the input device is 5 and using a /INCLUDE command where your card deck would normally go. The /INCLUDE command takes the form of /INCLUDE filename where filename is the name of your data file (in this case). See page 115 of the MUSIC at NTSU Manual for further information regarding /INCLUDE. (NOTE: It will be necessary for you to edit your data file to delete the two blank cards output by the card punch and to insert any end of data indicators that might be needed).

ADDITIONAL CAI AVAILABLE THROUGH MUSIC

The Computing Center has recently acquired TEACH, an author language and interpreter that was developed by Educational Computer Systems at Maricopa Community Colleges in Phoenix, Arizona. TEACH was developed to present drill and practice and to conduct instructional dialogues. In addition to the ability to author your own TEACH lessons, TEACH also comes with some lessons that were developed, using TEACH, by the Educational Computer Systems people at Maricopa. For complete documentation of TEACH, contact anyone on the Academic Computing staff at the Computing Center. In order to participate in the existing TEACH lessons, sign on to MUSIC and enter the keyword TEACH. TEACH will respond by asking you your first and last name and who your teacher is (NONE is an acceptable answer for the latter question). It will then ask you what lesson you want. Any of the following lessons may be chosen (TEACH works best on ASCII terminals - i.e., MIME, LA36, ACTIV, etc.).

TEACH Lessons on TEACH

TEACH1, TEACH2, TEACH3, TEACH4, TEACH5, TEACH6.

English Computer Assisted Instruction Programs

1. ENGTEST Tests the student on his understanding of various phases of English grammar. Based on the student's performance, it then recommends various other programs which will help him/her to improve those skills in which (s)he is weak.
2. ENG1 Describes the basic elements of a complete sentence and gives the student practice at recognizing these elements.
3. ENG2 Gives the student practice at recognizing fragments and complete sentences. It asks the student to identify what element(s) of a complete sentence is(are) missing from each fragment.
4. ENG3 Describes run-on sentences and comma splices. It gives the student practice at identifying run-ons, comma splices, and properly punctuated sentences.
5. ENG4 Discusses the possessive case and the use of the apostrophe in forming this case. It asks the student to identify in sentences those words which are in the possessive case and to punctuate them correctly.
6. ENG5 Discusses subject-verb agreement, with particular emphasis on tricky pronouns ("either-neither," "anyone," "everyone," etc.), compound subjects, subjects joined by "or," and the use of "there is" and "there are."
7. ENG6A Discusses the use of "lie" and "lay."
8. ENG6B Discusses "to-too-two" and "then-than."
9. ENG6C Discusses "their-there-they're" and "its-it's."
10. ENG6D Discusses "accept-except" and "affect-effect."
11. ENG6E Discusses "used to," "supposed to," "good-well," "real - really."
12. ENG7 Discusses pronoun case (nominative and objective) and gives the student practice at choosing the proper case.
13. ENG8 Discusses pronoun agreement (number and person) and gives the student practice at choosing the correct pronoun to agree with such troublesome antecedents as "people," "person," "one," etc.
14. ENG9 Defines and discusses adverb clauses and gives the student practice at identifying these clauses.
15. ENG10 Discusses the use of the apostrophe with contractions. It asks the student to distinguish between sentences which need contractions and those which need possessive pronouns.
16. ENG21 Defines "gerund" and "gerund phrase." It asks the student to identify sentences which use a gerund or gerund phrase as subject and to distinguish these sentences from others which do not use this construction.
17. ENG22 Defines a present participle and explains the use of participial phrases in a sentence. It asks the student to identify introductory and embedded participial phrases and to punctuate them correctly.
18. ENG23 Describes prepositional phrases and their punctuation, giving the student practice at recognizing and punctuating short prepositional phrases.
19. SPELL Presents commonly misspelled words randomly, prints over the word, and asks the student to spell the word just presented. There are currently 6 sets of words to choose

from.

TEACH Lessons on MUSIC Commands, Editor, and SCRIPT

Lessons available in Course #1 (does not mention /INPUT files):

1. MUSIC1 MUSIC immediate commands: /ID, /OFF, /LIST, /DISPLAY, /CANCEL, /EXEC, /LIB, /PURGE, HELP
2. MUSIC2 Additional immediate commands and utilities: /CTL, /TABIN, /PENAME, /SKIP, (@LIST, @BATCH, @VS1, @DCS - not currently available).
3. MUSIC3 MUSIC deferred commands: /DATA, /INCLUDE, /JOB, /LOAD, /OPT, /SYS, and /FILE
4. PRFL1 How to use the MUSIC PROFIL program to change various parameters in the user profile.
5. EDITOR1 Elementary Editor commands: =, NUM, TOP, LAST, NEXT, UP, LOCATE, SEARCH, QUIT, SAVE, LIST, PRINT, CHANGE, as well as /EDIT filename NEW and /EDIT filename.
6. EDITOR2 More Editor commands: INPUT, MERGE, FILE, HELP, INSEPT, REPLACE, DELETE, SCAN
7. EDITOR3 Still more Editor commands: MOVE, COPY, FIND, HUNT, DELIM, ADD, BLANK, OVERLAY, EXEC, and =/string1/string2/
8. MUSDIR List of lesson names/descriptions of Course #1 .

Lessons available in Course #2 parallel those in course #1 numbers 1-7, except that they cover the /INPUT file in their discussion. The lesson names are:

1. JCL1 2. JCL2 3. JCL3 4. PRFL 5. EDIT1 6. EDIT2 7. EDIT3
8. JCLDIR contains a list of lesson names/descriptions for Course #2.

Lessons Common to Both Course #1 and Course #2

1. SCRIPT1 Elementary SCRIPT commands: .FI, .NF, .RI, .LL, .IN, .DS, .SS, .SP, plus the /TEDIT command
2. SCRIPT2 Intermediate SCRIPT commands: .PL, .PA, .TT, .TM, .BT, .TB, .TS, .DK, .BR, .CE, .CM, .UN, plus .UL MASK

TEACH Tutorials on VS BASIC

1. VSB An introductory lesson which offers some optional definitions of Data Processing terms and a description of available lessons. This lesson can be skipped entirely if students already know the meaning of terms such as "hardware," "software," "file," etc.
2. VSB1 Introduces a simple VS BASIC program. It discusses the REM, PRINT, INPUT, IF..THEN, GO TO, and END statements as well as variable names. A quiz is offered at the end.
3. VSB2 Discusses the /BASIC Editor and how to create, change, and execute VS BASIC programs. A programming assignment based on VSB1 is offered at the end.
4. VSB3 Discusses LET statements, arithmetic, relational, and logical operators, and priorities of operations in evaluating expressions. A quiz is offered at the end.
5. VSB4 Discusses infinite loops, loops with incremented counters, and FOR-NEXT loops, with a quiz at the end.
6. VSB5 Discusses various PRINT features, including full and packed print zones, inserting variables in text, and the

- TAB function. A programming assignment based on VSB4/VSB5 is offered at the end.
7. VSB6 Continues the discussion of the PRINT statement with PRINT USING and FOPM. Print charts are also introduced. A quiz is offered at the end.
 8. VSB7 Introduces the READ, DATA, and PESTORE statements. A programming assignment based on VSB6/VSB7 is offered at the end.
 9. VSB8 Discusses creating, writing, and reading data files. OPEN, CLOSE, GET, PUT, and RESET statements, as well as the /FILE deferred command, are introduced, with a quiz at the end.
 10. VSB9 Presents material on GOSUB's, the STOP statement, and the DAT and STR functions. A programming assignment based on VSB8/VSB9 is offered at the end.
 11. VSB10 Discusses intrinsic and user-defined functions, including DEF, RETURN, and FNEND statements, and the string operator .CAT. A quiz is offered at the end.
 12. VSB11 Discusses arrays or matrices, subscripts, dimensioning, and some of the ways in which VS BASIC can manipulate arrays. Included are MAT READ, MAT PRINT, MAT INPUT, MAT GET, and MAT PUT. A quiz is offered at the end.
 13. VSB12 Discusses the use of tables in business data processing and presents a sample program which uses a table lookup. A quiz is offered at the end.
 14. VSB13 presents a simple sorting technique as well as material on the IDX function. A programming assignment is offered at the end.
 15. VSB14 Introduces computed GOTOs and GOSUBs, CHAIN and USE, and error handling with ON, EXIT, and error options on GET. A final exam is offered at the end.

Programming assignments available as separate TEACH lessons:

1. VSBA1 An assignment to create a program with a simple YES/NO question and response. It is based on VSB1.
2. VSBA2 An assignment to create two similar programs, one using a loop with an incremented counter and print zones, the other using FOR-NEXT and TAB. It is based on VSB4 and VSB5.
3. VSBA3 An assignment to do a payroll program with PEAD and DATA statements and a printer spacing chart. It is based on VSB6 and VSB7.
4. VSBA4 An assignment to do the same payroll program as in VSBA3, only this time using a data file and GOSUBs. It is based on VSB8 and VSB9 and uses data file PAYDAT.
5. VSBA5 An assignment to create a payroll program, using a table search for the employee's pay rate. It is based on VSB12 and uses data file PAYFILE.
6. FSBA6 An assignment to do a payroll program incorporating a sort routine. It is based on VSB13 and uses data files PAYTABLE and PAYSORT.

3 BASIC sample programs that may be executed, listed and examined from *GO mode:

1. VSB.DEMO1 A simple "ask a YES/NO question, evaluate the answer" program which can be used as a model for assignment VSBA1.
2. VSB.DEMO2A An infinite loop.
3. VSB.DEMO2B Loop with incremented counter.

1. VSB.DEMO2C FOR-NEXT loop.

The last two may be used as models for assignment VSBA2.

3. VSB.DEMO3 An inventory program using PHAD and DATA statements which may be used as a model for programming assignment VSBA3.
4. VSB.DEMO4 An inventory program using GOSUBs and a data file. This model for assignment VSBA4 uses data file INV DAT.
5. VSB.DEMO5 A program with a table lookup which can be used as a model for VSBA5. It uses data file DAT TAB.
6. VSB.DEMO6 An inventory program with a sort routine. It can be used as a model for VSBA6. It uses data files TAB6 and DEMOSORT

PLEASE NOTE: These programs have not been tested extensively and/or modified for NTSU other than making them run on our system. Because of this, the user may come into contact with references to the Maricopa system and/or instructions that do not apply to our system. Eventually, these (possible) inconsistencies will be corrected; however, in the meantime, it was felt that enough valuable information could be gained from these lessons to make it worthwhile for them to be available. Should any problems occur in using these lessons, please contact a member of the academic computing staff, either through the system mail facility or directly, through the Computing Center.

FUTURE MUSIC SOFTWARE

The Computing Center has recently acquired several programs from the Educational Computer Systems at Maricopa Community Colleges. They will be made available for general usage as they are installed and tested. One of the procedures that is currently being tested is TESTER, a computer assisted test construction program. Anyone who thinks that they might be interested in using TESTER should contact Tom Madron in the Computing Center.

An in-house program is also being developed that will enable the user to upload files from MUSIC to OS and download files from OS to MUSIC.

MUSIC NOTES

1. Please note an ****ERROR**** in the April BENCHMARKS with regard to the general ID available to non-MUSIC users for LEARNING purposes. It was erroneously reported that to sign-on to MUSIC one must simply type /LA00 ; actually one should type /ID LA00. A slight to major inconvenience will result when more than one user is signed on under this general ID and is trying to use the same IIS course at the same time. A message will be received which states, in effect, "course in use" . Unfortunately, the only solution to this problem is to sign-off and try again another time.
2. The January 1981 BENCHMARKS discussed, in detail, the management of SAVE library files. Remember that when you are going to search your library for a particular set of programs, an asterisk (*) must be placed at each level of the name that does not contain the identifier you are after. For example: If you have several COBOL programs that have three level names (i.e., separated by periods), you would enter LIB N(*.*.COB) if the identifier was in the last level of the name; if the identifier was at the beginning of the name, you would enter LIB N(COB.*.*).

3. When entering text to be processed using MUSIC/Script, a // entered beginning in column 1 (while in /INPUT mode) will cause the entire line of the text to be capitalized. The only solution to this seems to be to avoid entering a // beginning in column 1. This could be accomplished by either preceding the // with a blank or embedding the // in text by moving text from the previous line to start in column 1.
4. MUSIC backup is done daily, Monday through Friday at 4 a.m. ; Saturday and Sunday it is done at 11:30 p.m., just before the system is shut down.

```

* * * * *
*       H P - 2000       *
* * * * *

```

GETTING STARTED ON THE HP

The following text is adapted from a class handout prepared by Dr. Reynolds Griffith, Associate Professor of Finance, Insurance, Real Estate and Law:

The H-P terminal system is both useful and easy to use. It does not require any knowledge of programming in order to use it because there are a wide variety of stored programs available. (It is also easy to learn to program if you want to write your own programs) Use of the system requires an ID number and ID password, obtainable from the Computing Center (or your instructor, in a classroom situation).

Terminals are located in Room 153 of the Business Administration building, the keypunch room in the Computing Center, and various other locations around campus (see "Accessing the Interactive Computing Systems" in the March 1981 BENCHMARKS for a complete list of terminal locations). Some of the terminals are "hard copy" - that is, they type on paper. Others are "CRTs" - the typing appears on a screen.

LOG ON PROCEDURE

(Omit steps 2-4 for a "hardwired terminal")

1. Check terminal switches - the LOCAL-LINE switch should be on LINE, and BAUD switch on 300.
2. Dial the computer's telephone number: 2771 or 9-566-4736.
3. Turn the OFF-ON switch of the acoustical coupler to ON.
(This switch is located on the front of some couplers, on the back of others. The other switch on the back should be set to "full duplex.")
4. When you hear a shrill tone, plug the telephone handset into the acoustical coupler. (Note that the handset goes in frontwards on some couplers, backwards on others.)
5. Turn power OFF-ON switch on front of terminal to ON.
Press RETURN and LINEFEED keys. (Computer should type "PLEASE LOG IN").
7. Type HELLO-ID,PASSWORD for example: HELLO-B699,XYZABC
(ID number and comma are typed in regular way, password is typed while depressing Control key) and then depress the RETURN key. (Computer should type message, then "READY" after which you can begin.)

RUNNING A PROGRAM

Type GET-programname (e.g. GET-IDA), then type RUN on the next line (remembering that the RETURN key must be pressed after each line) or type EXE-programname.

If a program name is of the form \$PROGRAM, it means that it is in the system library. *PROGRAM means it is in the group library, PROGRAM means it is stored under the ID number being used. If you do not type the program name in proper form, the computer will tell you "NO SUCH PROGRAM", since it looks for it only in the location specified by the form. Many programs have instructions built into them to tell you how to use them. Instructions for others may be given to you by your instructor. Programs may expect answers to yes or no questions in the form of YES and NO, Y and N, or 0 and 1, and do not always tell you which to use. Remember that each time you type the answer to a question or input data, you must depress the RETURN key.

CORRECTING MISTAKES

You can correct mistakes before you depress the RETURN key. If only the last character or two which you have typed is wrong, you can eliminate them by depressing the BACKSPACE key (CONTROL key and H on some terminals) once for each consecutive character to be eliminated. (It doesn't physically wipe them out, but they're not fed into the computer). To eliminate a whole line, hold down the CONTROL key and depress X. Once you press the RETURN key, your mistake has been fed into the computer and it is usually impossible to correct it. However, you can stop a program by pressing the BREAK key and starting over. (You restart a program by typing RUN EXE-programname).

LOG-OFF PROCEDURE

1. Type BYE and depress the RETURN key.
2. Disconnect the telephone handset from the coupler and hang up the phone.
3. Turn the coupler OFF-ON switch to OFF. (If it is on the back of the coupler, be sure you don't turn the wrong switch).
4. Turn the terminal OFF-ON switch to OFF.
5. Tear your paper output off and clean up any debris.

PROGRAMS TO RUN

If you are enrolled in a class that will use the HP, your instructor will give you the names of programs to run for class purposes. Otherwise, you can familiarize yourself with the procedures by logging on and executing the NEWS program mentioned in the message the computer types. One way to become comfortable in using the terminals is to play some of the games stored in the system. Don't spend too much time on the games. Game players must always yield a terminal if a serious user needs it. If you want to learn how to program the H-P system, begin with the program named \$TUT01 (zero, one), which will teach you step by step. In a few hours you will be able to write useful programs of your own.

THE BIG PURGE

The HP system (as reported in the April BENCHMARKS) has been purged, as of June 1, 1981, of all programs and files not accessed since January 1, 1981. August 29, 1981, marks the deadline for the renewal of individual ID's. All individuals possessing HP ID's who have not submitted a renewal form by this date will have their programs and files purged on August 30, 1981. Renewal forms may be obtained from one of the secretaries in the reception area of the Computing Center (ISB 119).

ISB 224 NO LONGER A TERMINAL ROOM

Due to overcrowding and general lack of space, ISB 224 has been converted into office space for Computing Center personnel. The terminals that were located in that room have been redistributed to various locations throughout the campus. See the BENCHMARKS March 1981 issue "Accessing the Interactive Computing Systems" for a complete list of terminal locations.

SUMMER BACKUP SCHEDULE

The backup of the HP-2000 continues to follow the schedule established in the fall. A daily backup will be done between 8 a.m. and 8:30 a.m., Monday through Friday and a weekly backup will be performed every Friday between 4 and 5 p.m. .

SAS PROCEEDINGS AVAILABLE FOR EXAMINATION

The proceedings of the Sixth Annual SAS User's Group International Conference, held February 8-11, 1981, are available for examination, upon request, from Claudia Putnam. The proceedings contain articles and abstracts of articles by people from a wide variety of backgrounds on such topics as: Graphic Data Presentation; Statistics (GLM); Statistics (Non-GLM); Statistical Consulting; Training and Support; Management Information Systems; Research Information Systems; and Information Systems. Feel free to drop by and look through the table of contents. There are many articles that would probably benefit anyone who uses SAS extensively, especially those users of statistical procedures.

RESULTS OF THE WORKSHOP/SEMINAR POLL

The response to the Workshop/Seminar Checklist in the April BENCHMARKS was, to say the least, light, nevertheless, here are the results:

1. MUSIC
 - A. General Overview - 5
 - B. Working with Files - 3
 - C. Accessing the OS operating system - 4
 - D. Using the editors - 3
 - E. MUSIC/Script - 4
 - F. IIS - 4
 - G. Other - 0
2. HP-2000
 - A. General overview - 1
 - B. Working with files - 1
 - C. Other - 0
3. Statistical Packages
 - A. SPSS
 1. General overview - 3
 2. Other - 0
 - B. SAS
 1. General overview - 4
 2. Other: JCL and interpretation of output - 1
 - C. Other statistical packages
 1. Package: OSIRIS - 3

Presentation - 0

4. JOB Control Language (JCL)

A. General overview - 1

B. Specific topic(s) - 0

5. IBM Utilities- 0

6. Miscellaneous topics: CAD/Cam programs for industry - 1

What the computer can do for the researcher - 1

Please feel free to continue responding on any or all of the above topics, either through the electronic mail system, the intercampus mail system, or the United States postal system. The Computing Center staff associated with academic computing is already working to disseminate information on many of the topics listed above, although not necessarily in seminar form. Upcoming issues of BENCHMARKS will continue to notify users and potential users of methods of self-education that become available, whether through computer assisted instruction (CAI), workshops, seminars, slide presentations, publications, or combinations of these.

INDEX TO PAST ISSUES

In order to utilize BENCHMARKS to its fullest capacity and avoid redundancies, an index of previous issues containing information considered still pertinent to the NTSU Computing Center is included in each issue.

<u>SUBJECT</u>	<u>VOL/NO</u>	<u>MONTH/YEAR</u>	<u>PG</u>
1. ACADEMIC COMPUTING IN GENERAL			
Planning for Academic Computing	2/2	March/1981	1
ICPSR - A University Wide Resource	2/2	March/1981	7
2. ACCESS TO COMPUTING FACILITIES			
Accessing the Interactive			
Computing Systems	2/2	March/1981	5
So You Want to Compute . . .	1/3	October/1980	2
3. HARDWARE			
General Terminal Features	2/3	April/1981	6
Telex Customer Phone Number	2/3	April/1981	8
Take-Home Terminals Now			
Available For Faculty Use	2/2	March/1981	7
Terminal Maintenance			
Contract	2/1	January/1981	7
The Computing Center Goes			
Through Some Changes	1/3	October/1980	1
NTSU Computing Center General			
Usage Equipment List	1/3	October/1980	5
The New 3330/3350 Compatible Disk			
Drives - How "The Change" Can			
Effect You	1/3	October/1980	6
News From College of Business			
Computing Center	1/3	October/1980	12
Computer Acquisition			
Status Report	1/2	June/1980	1
4. INPUT			
Job Card Preparation	2/3	April/1981	2
Disk Data Set Naming Conventions	2/2	March/1981	2
Memory No Longer A Factor In Job			
Class Selection	1/3	October/1980	10

<u>SUBJECT</u>	<u>VOL/NO</u>	<u>MONTH/YEAR</u>	<u>PG</u>
5. OUTPUT			
Job Processing Schedule	2/3	April/1981	4
Routing Jobs from OSJR	2/2	March/1981	3
High Speed Print/Punch in Music	2/2	March/1981	3
Secure Job Pickup	2/2	March/1981	6
Computing Center RJE			
Means Faster Turnaround			
For Most Users	2/1	January/1981	2
6. SERVICES AVAILABLE			
Services Available to Users of			
The NTSU Computing Facilities	1/3	October/1980	2
Data Entry Services	1/3	October/1980	10
7. SOFTWARE			
Learning To Use MUSIC	2/3	April/1981	5
MUSIC Notes	2/3	April/1981	7
Alternative SPSS Procedure	2/3	April/1981	3
The Type I, II, III,			
and IV Dilemma	3/2	April/1981	11
MUSIC Notes	2/2	March/1981	4
ETS	2/2	March/1981	6
GALILEO IV	2/2	March/1981	7
Program Products Available			
For MVT	2/1	January/81	3
Status of MUSIC, Spring 1981	2/1	January/81	4
OSJE/OSJR	2/1	January/81	5
Management Of MUSIC SAVE			
Library Files	2/1	January/81	6
ANOVA Models Tested by Various			
SPSS and SAS Options	2/1	January/81	10
Software - What We Have and			
What We Support	1/3	October/1980	3
Interactive Computing On the AS/5000			
Through MUSIC	1/3	October/1980	9
SAS Supplemental Library	1/3	October/1980	11
VM/370	1/2	June/1980	7
Stat Users Beware	1/2	June/1980	10
SPSS	1/2	June/1980	11
Behavioral and Social Science Data			
Analysis Using Osiris	1/2	June/1980	13
8. USAGE STATISTICS			
HP/2000 Usage Statistics-September 3,			
1979 to September 1, 1980	1/3	October/1980	7
AS/5000 Usage Statistics-September 1,			
1979 to August 31, 1980	1/3	October/1980	8
9. UTILITY APPLICATIONS			
Changing Dataset Names With			
IEHPRGM	2/3	April/1981	5
Deleting Disk Files	2/2	March/1981	3

CALLING ALL COURSES USING STATISTICAL PACKAGES

It has become apparent that there is a need for some type of list of courses that deal with the various statistical packages. There are many potential students for these classes, however they do not know where to go to find the class they desire. BENCHMARKS would like to help these people in their quest, but in order to do this, we must enlist the help of the professors, instructors, etc. who are offering the courses. So ... if you are planning to teach a class in the next year that will be focused around one or more of the statistical packages, please take a moment to provide the following information. (If you do not want to fool with the paper work, you can send a message to the "SYSTEM" while logged on to MUSIC - see "Electronic Mail System Revisited" in this issue of BENCHMARKS for more information on sending mail):

1. Course Title (if known) _____
2. School/Department course taught in _____
3. Short description of course _____
4. Semester(s) course will be taught _____
5. Prerequisites for attending course _____
6. Other pertinent information _____


```

* * * * *
*
* BENCHMARKS is a publication of the NTSU Computing Center.
* Reader/user feedback is encouraged, so send all letters,
* suggestions, etc., to: NTSU Computing Center
*                        NT Box 13495
*                        Denton, Tx 76203
*
* Claudia Putnam . . . . . BENCHMARKS Editor
*
* Richard Harris . . . . . Director of
*                        Computer Systems
*
* Thomas Madron . . . . . Manager, Academic
*                        Computing Services
* * * * *

```

BENCHMARKS is the Computing Center newsletter aimed at the NTSU academic community. A benchmark is a point of reference from which measurements of any sort are made, and that name was chosen to symbolize the intent of this publication - to provide a point of reference for the users of the North Texas State University computing facilities. BENCHMARKS was created to provide a channel of communications between the NTSU Computing Center and its users - a channel, hopefully, that will flow both ways, enabling both the users and the Computing Center staff to gain new insights into the ever changing world of academic computing.

PLEASE RETURN TO:

Academic Computing Services
The Computing Center
NT Box 13495
North Texas State University
Denton, TX 76203